Permutation classes: structure and combinatorial properties

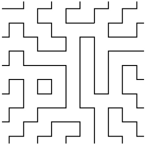
Mathilde Bouvel

Institut für Mathematik, Universität Zürich

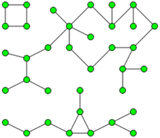
December 17, 2013.

Enumerative combinatorics

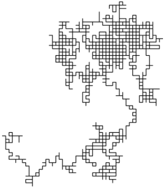
- Examples of discrete objects:



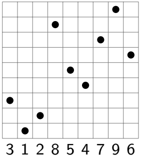
Fully-Packed Loops



Graphs



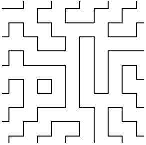
Walks



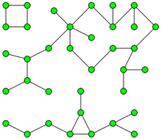
Permutations

Discrete objects and combinatorial classes

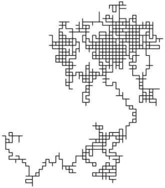
- Examples of discrete objects:



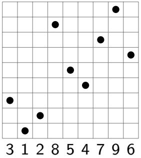
Fully-Packed Loops



Graphs



Walks

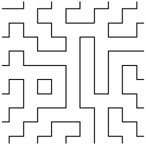


Permutations

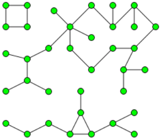
- A **combinatorial class** is a family of discrete objects is equipped with a notion of size such that for every integer n, the set of objects of size n is finite.

Discrete objects and combinatorial classes

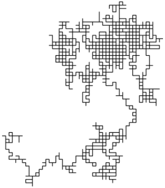
- Examples of discrete objects:



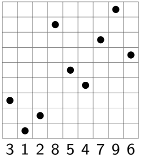
Fully-Packed Loops
$n = 10$



Graphs
$n = 36$



Walks
$n = 2500$

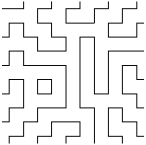


Permutations
$n = 9$

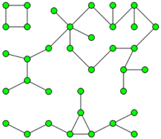
- A **combinatorial class** is a family of discrete objects is equipped with a notion of size such that for every integer n, the set of objects of size n is finite.

Discrete objects and combinatorial classes

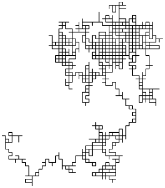
- Examples of discrete objects:



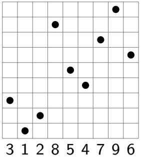
Fully-Packed Loops
$n = 10$



Graphs
$n = 36$



Walks
$n = 2500$



Permutations
$n = 9$

- A **combinatorial class** is a family of discrete objects is equipped with a notion of size such that for every integer n, the set of objects of size n is finite.

- Characterize the objects in a combinatorial class and study their combinatorial properties.

This may help understanding phenomena modeled by discrete objects.

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

- Simplest question: How many objects of size n are there in $\mathcal{C}$?

Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)

$\hookrightarrow$ $Cat_n = \frac{1}{n+1}\binom{2n}{n}$, $Bax_n = \sum_{k=1}^{n} \frac{2}{n(n+1)^2}\binom{n+1}{k-1}\binom{n+1}{k}\binom{n+1}{k+1}$

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$

$\hookrightarrow$ $c_{n,k}$ = number of objects of size n with value of the parameter k

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$
- Asymptotic equivalent for c_n

$\hookrightarrow$ $Si_n \sim \frac{n!}{e^2}$, but not exact formula for Si_n

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$
- Asymptotic equivalent for c_n
- Equi-enumeration of several classes (preserving the distribution of some statistics)

$\hookrightarrow$ Proved computationally or with size-preserving bijections.

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$
- Asymptotic equivalent for c_n
- Equi-enumeration of several classes (preserving the distribution of some statistics)
- Explicit or implicit expression of the **generating function** $C(z) = \sum_n c_n z^n$

$\hookrightarrow$ $Cat(z) = \frac{1-\sqrt{1-4z}}{2z}$

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$
- Asymptotic equivalent for c_n
- Equi-enumeration of several classes (preserving the distribution of some statistics)
- Explicit or implicit expression of the **generating function** $C(z) = \sum_n c_n z^n$
- Nature of the generating function (rational, algebraic,...)

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$
- Asymptotic equivalent for c_n
- Equi-enumeration of several classes (preserving the distribution of some statistics)
- Explicit or implicit expression of the **generating function** $C(z) = \sum_n c_n z^n$
- Nature of the generating function (rational, algebraic,...)
- Bivariate generating functions $C(z;u) = \sum_n c_{n,k} u^k z^n$

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Exact formula for c_n (closed form, or summation)
- Enumeration refined with some statistics, $c_n = \sum_k c_{n,k}$
- Asymptotic equivalent for c_n
- Equi-enumeration of several classes (preserving the distribution of some statistics)
- Explicit or implicit expression of the **generating function** $C(z) = \sum_n c_n z^n$
- Nature of the generating function (rational, algebraic,...)
- Bivariate generating functions $C(z;u) = \sum_n c_{n,k} u^k z^n$
- Asymptotic behavior of $C(z)$ near the singularity

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

• Simplest question: How many objects of size n are there in $\mathcal{C}$?
Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

- Many ways of answering this question!

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

- Simplest question: How many objects of size n are there in $\mathcal{C}$?

Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

  - Many ways of answering this question!

- Further questions:
  - Combinatorial characterization or properties of the objects

Such questions are often answered in the proof of an enumeration result.

What do we want to know about combinatorial classes?

Let $\mathcal{C}$ be a combinatorial class.

- Simplest question: How many objects of size n are there in $\mathcal{C}$?

Let $\mathcal{C}_n$ be the set of objects of size n in $\mathcal{C}$, and $c_n = |\mathcal{C}_n|$.

  - Many ways of answering this question!

- Further questions:
  - Combinatorial characterization or properties of the objects

Such questions are often answered in the proof of an enumeration result.

- More general approach:
  - Define frameworks where all combinatorial classes have common properties

$\hookrightarrow$ For all simple varieties of trees, $c_n \sim \gamma \rho^{-n} n^{-3/2}$.

Permutations

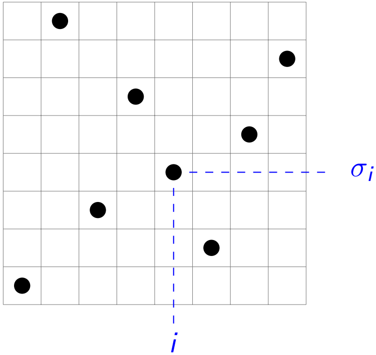
Permutation of size n = Bijection from $[1..n]$ to itself.
Set $\mathfrak{S}_n$, and $\mathfrak{S} = \cup_n \mathfrak{S}_n$.

Permutations

Permutation of size n = Bijection from $[1..n]$ to itself.
Set $\mathfrak{S}_n$, and $\mathfrak{S} = \cup_n \mathfrak{S}_n$.

- Two lines notation:
  $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 8 & 3 & 6 & 4 & 2 & 5 & 7 \end{pmatrix}$
- Linear notation:
  $\sigma = 1\ 8\ 3\ 6\ 4\ 2\ 5\ 7$
- Description as a product of cycles:
  $\sigma = (1)\ (2\ 8\ 7\ 5\ 4\ 6)\ (3)$

- Graphical description, or diagram:



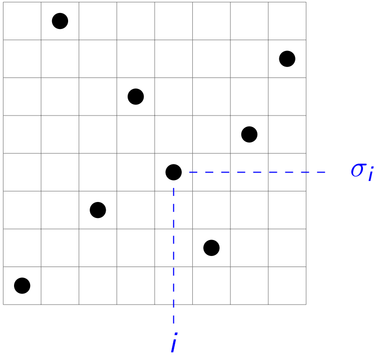
Permutation of size n = Bijection from $[1..n]$ to itself.
Set $\mathfrak{S}_n$, and $\mathfrak{S} = \cup_n \mathfrak{S}_n$.

- Two lines notation:
  $$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 8 & 3 & 6 & 4 & 2 & 5 & 7 \end{pmatrix}$$

- Linear notation:
  $\sigma = 1\ 8\ 3\ 6\ 4\ 2\ 5\ 7$

- Description as a product of cycles:
  $\sigma = (1)\ (2\ 8\ 7\ 5\ 4\ 6)\ (3)$

- Graphical description, or diagram:

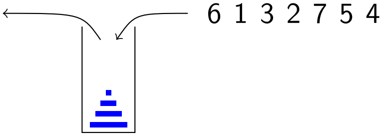


More precisely, I study **permutation patterns** and **permutation classes**.

Permutation patterns and permutation classes

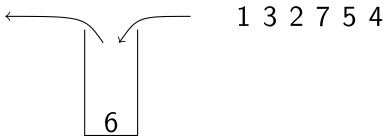
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



The stack sorting operator S

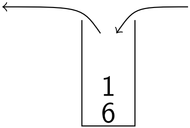
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.

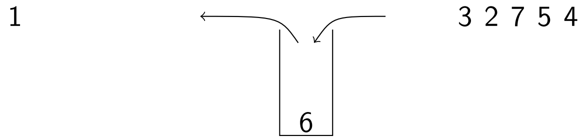


3 2 7 5 4

The origin of permutation patterns: Stack sorting

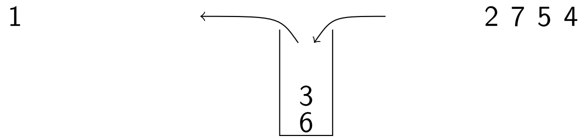
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



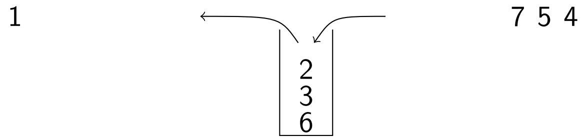
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



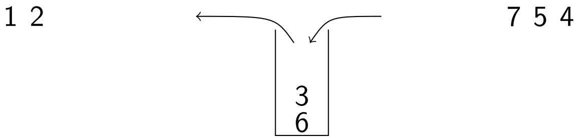
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



The stack sorting operator S

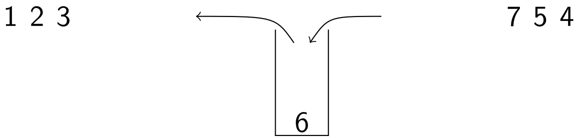
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

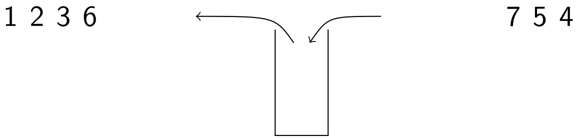
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



The stack sorting operator S

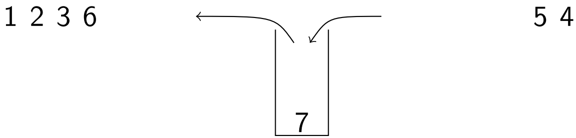
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

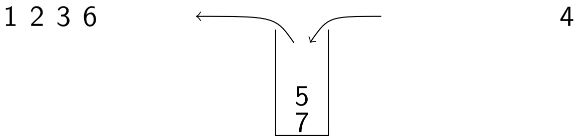
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



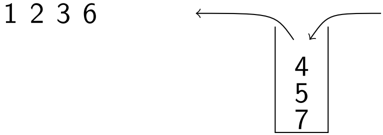
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



The stack sorting operator S

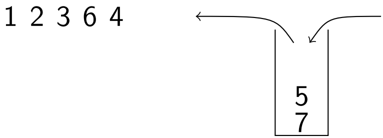
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

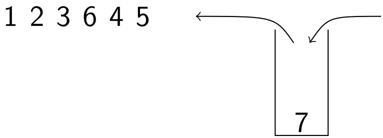
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

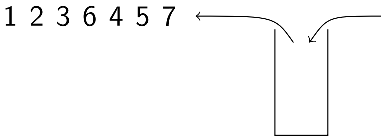
The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.



The stack sorting operator S

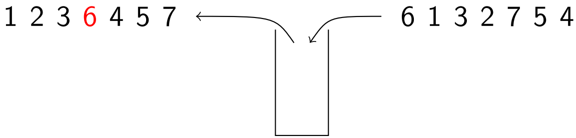
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

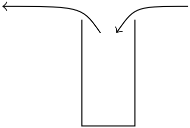
Sort (or try to do so) using a stack satisfying the Hanoi condition.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.

$\mathbf{S}(\sigma) =$ 1 2 3 6 4 5 7 ⟵  ⟵ 6 1 3 2 7 5 4 $= \sigma$

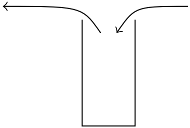
Equivalently, $\mathbf{S}(\varepsilon) = \varepsilon$ and $\mathbf{S}(LnR) = \mathbf{S}(L)\mathbf{S}(R)n$, $n = \max(LnR)$

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.

$\mathbf{S}(\sigma) =$ 1 2 3 6 4 5 7 ← ← 6 1 3 2 7 5 4 $= \sigma$

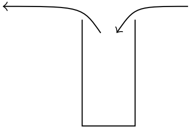


Equivalently, $\mathbf{S}(\varepsilon) = \varepsilon$ and $\mathbf{S}(LnR) = \mathbf{S}(L)\mathbf{S}(R)n$, $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :
A permutation σ is stack-sortable iff σ avoids the pattern 231

The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.

$\mathbf{S}(\sigma) = 1\ 2\ 3\ 6\ 4\ 5\ 7$  $6\ 1\ 3\ 2\ 7\ 5\ 4 = \sigma$

Equivalently, $\mathbf{S}(\varepsilon) = \varepsilon$ and $\mathbf{S}(LnR) = \mathbf{S}(L)\mathbf{S}(R)n$, $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ avoids the pattern 231

meaning that there are no $i < j < k$ such that $\sigma_k < \sigma_i < \sigma_j$,

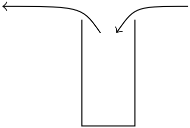
or equivalently no subsequences $\cdots \sigma_i \cdots \sigma_j \cdots \sigma_k \cdots$ of σ whose elements are in the same relative order as 231.

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.

$\mathbf{S}(\sigma) =$ 1 2 3 6 4 5 7



6 1 3 2 7 5 4 $= \sigma$

Equivalently, $\mathbf{S}(\varepsilon) = \varepsilon$ and $\mathbf{S}(LnR) = \mathbf{S}(L)\mathbf{S}(R)n$, $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ avoids the pattern 231

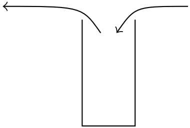
meaning that there are no $i < j < k$ such that $\sigma_k < \sigma_i < \sigma_j$,

or equivalently no subsequences $\cdots \sigma_i \cdots \sigma_j \cdots \sigma_k \cdots$ of σ whose elements are in the same relative order as 231.

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a stack satisfying the Hanoi condition.

$\mathbf{S}(\sigma) =$ 1 2 3 6 4 5 7  6 1 3 2 7 5 4 $= \sigma$

Equivalently, $\mathbf{S}(\varepsilon) = \varepsilon$ and $\mathbf{S}(LnR) = \mathbf{S}(L)\mathbf{S}(R)n$, $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ avoids the pattern 231

meaning that there are no $i < j < k$ such that $\sigma_k < \sigma_i < \sigma_j$,

or equivalently no subsequences $\cdots \sigma_i \cdots \sigma_j \cdots \sigma_k \cdots$ of σ whose elements are in the same relative order as 231.

Enumeration by the Catalan numbers $Cat_n = \frac{1}{n+1}\binom{2n}{n}$

Permutation patterns

Pattern relation $\preccurlyeq$:
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is in the same relative order ($\equiv$) as π.

Notation: $\pi \preccurlyeq \sigma$.

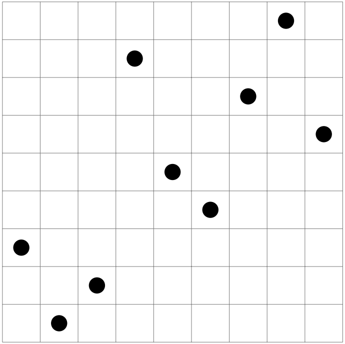
Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.

Example: $2\,1\,3\,4 \preccurlyeq \mathbf{3}\,\mathbf{1}\,2\,8\,\mathbf{5}\,4\,\mathbf{7}\,9\,6$
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.

Permutation patterns

Pattern relation $\preccurlyeq$:

$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is in the same relative order ($\equiv$) as π.



Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.

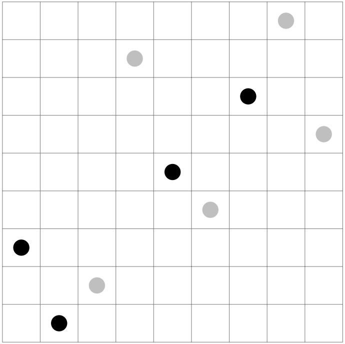
Example: $2\,1\,3\,4 \preccurlyeq$ **3 1** 2 8 **5** 4 **7** 9 6
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.

Pattern relation $\preccurlyeq$:
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is in the same relative order ($\equiv$) as π.

Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.

Example: $2\,1\,3\,4 \preccurlyeq$ **3 1** 2 8 **5** 4 **7** 9 6
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.



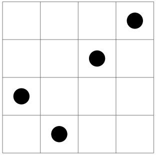
Permutation patterns

Pattern relation $\preccurlyeq$:
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is in the same relative order ($\equiv$) as π.

Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.

Example: $2\,1\,3\,4 \preccurlyeq \mathbf{3}\,\mathbf{1}\,2\,8\,\mathbf{5}\,4\,\mathbf{7}\,9\,6$
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.



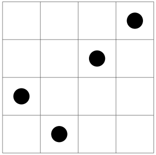
Permutation patterns

Pattern relation $\preccurlyeq$:

$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is in the same relative order ($\equiv$) as π.

Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.



Example: $2\,1\,3\,4 \preccurlyeq \mathbf{3}\,\mathbf{1}\,2\,8\,\mathbf{5}\,4\,\mathbf{7}\,9\,6$
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.

Remark: $\preccurlyeq$ is a partial order on $\mathfrak{S} = \bigcup_n \mathfrak{S}_n$.

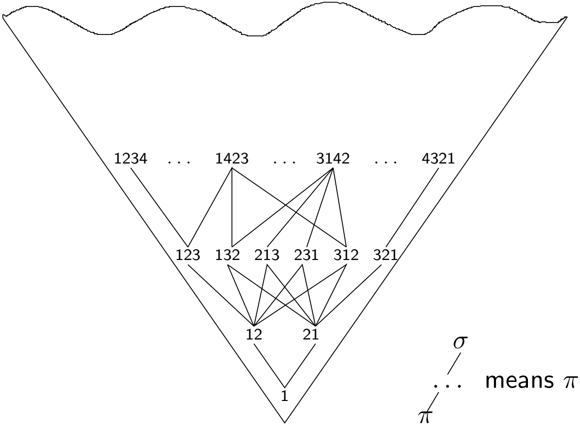
This is the key to defining permutation classes.

Permutation classes

• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.

Permutation classes

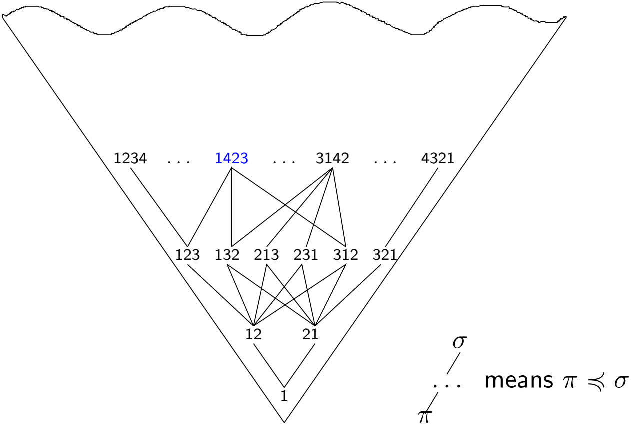
• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



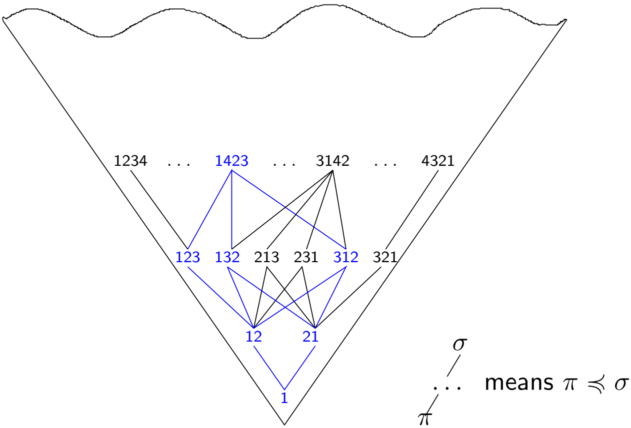
means $\pi \preccurlyeq \sigma$

Permutation classes

• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.

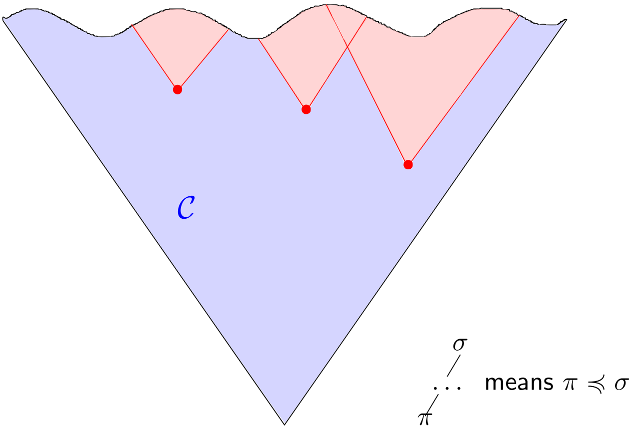


• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



Permutation classes

• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



Permutation classes

• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.

• Notations: $Av(\pi)$ = the set of permutations that avoid the pattern π

$$Av(B) = \bigcap_{\pi \in B} Av(\pi)$$

• Fact: For every permutation class $\mathcal{C}$, $\mathcal{C} = Av(B)$ for

$B = \{\sigma \notin \mathcal{C} : \forall \pi \preccurlyeq \sigma \text{ such that } \pi \neq \sigma, \pi \in \mathcal{C}\}$.

B is an antichain, called the **basis** of $\mathcal{C}$.

Permutation classes

• A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.

• Notations: $Av(\pi)$ = the set of permutations that avoid the pattern π

$$Av(B) = \bigcap_{\pi \in B} Av(\pi)$$

• Fact: For every permutation class $\mathcal{C}$, $\mathcal{C} = Av(B)$ for

$B = \{\sigma \notin \mathcal{C} : \forall \pi \preccurlyeq \sigma$ such that $\pi \neq \sigma, \pi \in \mathcal{C}\}$.

B is an antichain, called the **basis** of $\mathcal{C}$.

• Remarks:

- Conversely, every set $Av(B)$ is a permutation class.
- There exist infinite antichains, hence some permutation classes have infinite basis.

Permutation classes

- A **permutation class** is a set $\mathcal{C}$ of permutations that is downward closed for $\preccurlyeq$, i.e. whenever $\pi \preccurlyeq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.

- Notations: $Av(\pi)$ = the set of permutations that avoid the pattern π
  $Av(B) = \bigcap_{\pi \in B} Av(\pi)$

- Fact: For every permutation class $\mathcal{C}$, $\mathcal{C} = Av(B)$ for
  $B = \{\sigma \notin \mathcal{C} : \forall \pi \preccurlyeq \sigma \text{ such that } \pi \neq \sigma, \pi \in \mathcal{C}\}$.
  B is an antichain, called the **basis** of $\mathcal{C}$.

- Remarks:
  - Conversely, every set $Av(B)$ is a permutation class.
  - There exist infinite antichains, hence some permutation classes have infinite basis.

- Most results about permutation classes are enumeration results.

Some early specific enumeration results

- One excluded pattern:
  - of size 3:
    - Description of $Av(123)$ [MacMahon 1915] and $Av(231)$ [Knuth 68].
    - Enumeration by the Catalan numbers in both cases.
    - Bijections: [Simion, Schmidt 1985] [Claesson, Kitaev 2008].
    - For symmetry reasons, it is enough to study $Av(123)$ and $Av(231)$.

Some early specific enumeration results

- One excluded pattern:
  - of size 3:
    - Description of $Av(123)$ [MacMahon 1915] and $Av(231)$ [Knuth 68].
    - Enumeration by the Catalan numbers in both cases.
    - Bijections: [Simion, Schmidt 1985] [Claesson, Kitaev 2008].
    - For symmetry reasons, it is enough to study $Av(123)$ and $Av(231)$.
  - of size 4:
    Only three different enumerations. Representatives are:
    - $Av(1342)$ [Bóna 97], algebraic generating function
    - $Av(1234)$ [Gessel 90], holonomic (or D-finite) generating function
    - $Av(1324)$ … remains an open problem

Some early specific enumeration results

- One excluded pattern:
  - of size 3:
    - Description of $Av(123)$ [MacMahon 1915] and $Av(231)$ [Knuth 68].
    - Enumeration by the Catalan numbers in both cases.
    - Bijections: [Simion, Schmidt 1985] [Claesson, Kitaev 2008].
    - For symmetry reasons, it is enough to study $Av(123)$ and $Av(231)$.
  - of size 4:
    Only three different enumerations. Representatives are:
    - $Av(1342)$ [Bóna 97], algebraic generating function
    - $Av(1234)$ [Gessel 90], holonomic (or D-finite) generating function
    - $Av(1324)$ ... remains an open problem

- Systematic enumeration of $Av(B)$ when B contains small excluded patterns (size 3 or 4)

[Simion&Schmidt, Gessel, Bóna, Gire, Guibert, Stankova, West... in the nineties]

[Albert, Atkinson, Brignall, Callan, Kremer, Pantone, Shiu, Vatter, ... nowadays]

A general enumeration result for permutation classes

First common property of all (proper) permutation classes
(*i.e.* classes $\mathcal{C} \neq \mathfrak{S}$):

Theorem:

For every permutation π, $\sqrt[n]{|Av_n(\pi)|}$ converges to a constant c_π.

Conjectured by [Stanley & Wilf 92]; proved by [Marcus & Tardos 04].

A general enumeration result for permutation classes

First common property of all (proper) permutation classes
(*i.e.* classes $\mathcal{C} \neq \mathfrak{S}$):

Theorem:

For every permutation π, $\sqrt[n]{|Av_n(\pi)|}$ converges to a constant c_π.

Conjectured by [Stanley & Wilf 92]; proved by [Marcus & Tardos 04].

Hence, $|Av_n(\pi)| \approx c_\pi^n$.
To be compared with $|\mathfrak{S}_n| = n! \approx e^{n \log n}$.

A general enumeration result for permutation classes

First common property of all (proper) permutation classes
(*i.e.* classes $\mathcal{C} \neq \mathfrak{S}$):

Theorem:

For every permutation π, $\sqrt[n]{|Av_n(\pi)|}$ converges to a constant c_π.

Conjectured by [Stanley & Wilf 92]; proved by [Marcus & Tardos 04].

Hence, $|Av_n(\pi)| \approx c_\pi^n$.
To be compared with $|\mathfrak{S}_n| = n! \approx e^{n \log n}$.

This allows to define the **growth rate** of a class $\mathcal{C} = \limsup_{n \to \infty} \sqrt[n]{|\mathcal{C}_n|}$.

Consequence: All (proper) permutation classes have finite growth rates.

Except when $\mathcal{C} = Av(\pi)$, it is an open question to know
if $\lim_{n \to \infty} \sqrt[n]{|\mathcal{C}_n|}$ exists.

The general and the specific perspective

- Study of specific permutation classes
  - Characterization and enumeration, often with *ad hoc* arguments
  - Very precise results (distribution of statistics, . . .)
- Common properties of many permutation classes
  - Less precise results, but widely applicable

The general and the specific perspective

- Study of specific permutation classes
  - Characterization and enumeration, often with *ad hoc* arguments
  - Very precise results (distribution of statistics, ...)
- Common properties of many permutation classes
  - Less precise results, but widely applicable

Combinatorial properties or nice enumeration follow from nice structure.

- *Ad hoc* description of structure gives specific results.
- General notions of structure yield both specific and general results.

The general and the specific perspective

- Study of specific permutation classes
  - Characterization and enumeration, often with *ad hoc* arguments
  - Very precise results (distribution of statistics, ...)
- Common properties of many permutation classes
  - Less precise results, but widely applicable

Combinatorial properties or nice enumeration follow from nice structure.

- *Ad hoc* description of structure gives specific results.
- General notions of structure yield both specific and general results.

- Three possible looks at the structure of permutations:
  - Graphical structure, on the diagrams
  - Structure from substitution decomposition, with trees
  - Structure inherited from graphs

Structure from graphics

The first example: stack-sortable permutations

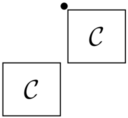
- Denoting $\mathcal{C}$ the class of stack-sortable permutations, we have [Knuth]:
  - $\mathcal{C} = Av(231)$;
  - $\mathcal{C}$ is enumerated by the Catalan numbers $Cat_n = \frac{1}{n+1}\binom{2n}{n}$.

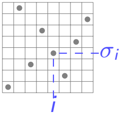
The first example: stack-sortable permutations

- Denoting $\mathcal{C}$ the class of stack-sortable permutations, we have [Knuth]:
  - $\mathcal{C} = Av(231)$;
  - $\mathcal{C}$ is enumerated by the Catalan numbers $Cat_n = \frac{1}{n+1}\binom{2n}{n}$.

- Proof: from the structure of $\mathcal{C}$.

Representing permutations as diagrams, we have

$\mathcal{C} = \varepsilon +$ 

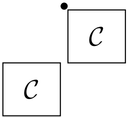
Remember diagrams: 

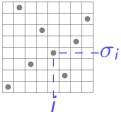
The first example: stack-sortable permutations

- Denoting $\mathcal{C}$ the class of stack-sortable permutations, we have [Knuth]:
  - $\mathcal{C} = Av(231)$;
  - $\mathcal{C}$ is enumerated by the Catalan numbers $Cat_n = \frac{1}{n+1}\binom{2n}{n}$.

- Proof: from the structure of $\mathcal{C}$.

Representing permutations as diagrams, we have

$\mathcal{C} = \varepsilon +$ 

Remember diagrams: 

- This recursive description characterizes $Av(231)$.
- It implies that $C(z) = 1 + zC(z)^2$,
  whose solution is the generating function $\sum_n Cat_n z^n$

Grid classes: the block structure of permutations

- Formalize the idea of describing permutation classes by "blocks".
  - A grid class $\mathcal{C}$ is defined by a matrix, like

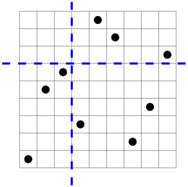
$$M = \begin{pmatrix} Av(1) & Av(12) \\ Av(21) & Av(132) \end{pmatrix} .$$

  - A permutation belongs to $\mathcal{C}$ if its diagram can be decomposed in rectangular blocks to fit into

$Av(1)$	$Av(12)$
$Av(21)$	$Av(132)$

.

Example:
$\sigma = 156398247 \in \mathcal{C}$ defined by M, because the diagram of σ can be decomposed as



Grid classes: the block structure of permutations

- Formalize the idea of describing permutation classes by "blocks".
  - A grid class $\mathcal{C}$ is defined by a matrix, like

$$M = \begin{pmatrix} Av(1) & Av(12) \\ Av(21) & Av(132) \end{pmatrix}.$$

  - A permutation belongs to $\mathcal{C}$ if its diagram can be decomposed in rectangular blocks to fit into

$Av(1)$	$Av(12)$
$Av(21)$	$Av(132)$

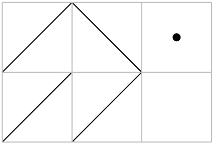
.

- Some results, applicable to grid classes $\mathcal{C}$ with additional restrictions:
  - $(\mathcal{C}, \preccurlyeq)$ is a partial well order (no infinite antichains)
  - $\mathcal{C}$ has a rational generating function
  - $\mathcal{C}$ has a finite basis

[Albert, Atkinson, Brignall, Ruškuc, Vatter, Waton]

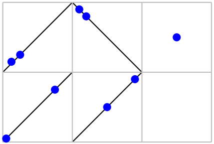
Even more geometry: Geometric grid classes

- A geometric grid class $\mathcal{C}$ is defined by a geometric figure F, like

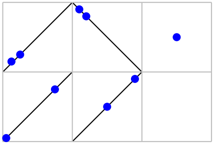


Even more geometry: Geometric grid classes

- A geometric grid class $\mathcal{C}$ is defined by a geometric figure F, like
- Permutations in $\mathcal{C}$ are those whose diagram that may be drawn on F (up to stretching and shrinking the axes)



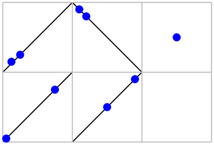
- A geometric grid class $\mathcal{C}$ is defined by a geometric figure F, like
- Permutations in $\mathcal{C}$ are those whose diagram that may be drawn on F (up to stretching and shrinking the axes)



• Remark: Many of the previously studied grid classes are equivalently described as geometric grid classes.

Even more geometry: Geometric grid classes

- A geometric grid class $\mathcal{C}$ is defined by a geometric figure F, like
- Permutations in $\mathcal{C}$ are those whose diagram that may be drawn on F (up to stretching and shrinking the axes)



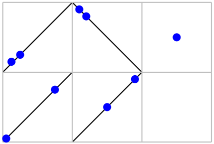
• Remark: Many of the previously studied grid classes are equivalently described as geometric grid classes.

• Theorem [Albert, Atkinson, Bouvel, Ruškuc & Vatter 13]:
Every subclass of a geometric grid class

- is a partially well ordered for $\preccurlyeq$;
- has a finite basis;
- is in size-preserving bijection with the words of a regular language;
- hence has a rational generating function.

Even more geometry: Geometric grid classes

- A geometric grid class $\mathcal{C}$ is defined by a geometric figure F, like
- Permutations in $\mathcal{C}$ are those whose diagram that may be drawn on F (up to stretching and shrinking the axes)



• Remark: Many of the previously studied grid classes are equivalently described as geometric grid classes.

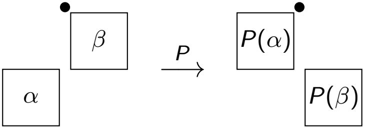
• Theorem [Albert, Atkinson, Bouvel, Ruškuc & Vatter 13]:
Every subclass of a geometric grid class

- is a partially well ordered for $\preccurlyeq$;
- has a finite basis;
- is in size-preserving bijection with the words of a regular language;
- hence has a rational generating function.

• Also, use of geometric grid classes in many specific recent works.

A graphical bijection preserving structure

- $P : Av(231) \to Av(132)$ defined recursively as follows:

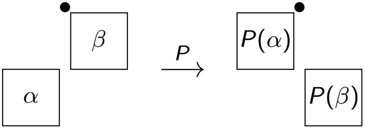


, with $\alpha, \beta \in Av(231)$

is a bijection between $Av(231)$ and $Av(132)$.

A graphical bijection preserving structure

- $P : Av(231) \to Av(132)$ defined recursively as follows:



, with $\alpha, \beta \in Av(231)$

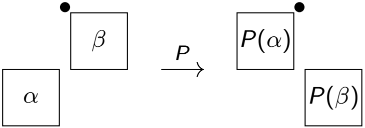
is a bijection between $Av(231)$ and $Av(132)$.

- It preserves the join distribution of many classical permutation statistics:
  - number and positions of the right-to-left maxima
  - number and positions of the left-to-right maxima
  - up-down word
  - and hence all statistics determined by the up-down word (in particular the descent set)

[Dokos, Dwyer, Johnson, Sagan & Selsor 12, Albert & Bouvel 13]

A graphical bijection preserving structure

- $P : Av(231) \to Av(132)$ defined recursively as follows:



, with $\alpha, \beta \in Av(231)$

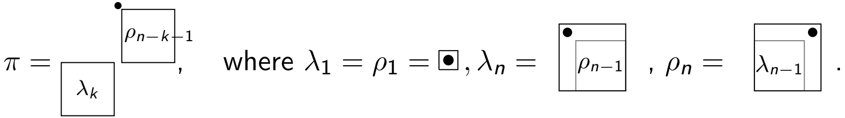
is a bijection between $Av(231)$ and $Av(132)$.

- It preserves the join distribution of many classical permutation statistics:
  - number and positions of the right-to-left maxima
  - number and positions of the left-to-right maxima
  - up-down word
  - and hence all statistics determined by the up-down word (in particular the descent set)

[Dokos, Dwyer, Johnson, Sagan & Selsor 12, Albert & Bouvel 13]

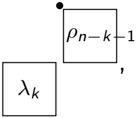
This graphically-guided bijection may be used to prove two general results.

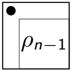
- Theorem [Albert & Bouvel 13]:
  - A pattern $\pi \in Av(231)$ is such that P provides a bijection between $Av(231, \pi)$ and $Av(132, P(\pi))$ if and only if

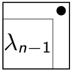


  - In particular, for such π, $Av(231, \pi)$ and $Av(132, P(\pi))$ are Wilf-equivalent, *i.e.* have the same enumeration.

- Theorem [Albert & Bouvel 13]:
  - A pattern $\pi \in Av(231)$ is such that P provides a bijection between $Av(231, \pi)$ and $Av(132, P(\pi))$ if and only if

$\pi =$ 

, where $\lambda_1 = \rho_1 = $ , $\lambda_n =$ 

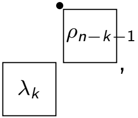
, $\rho_n =$ 

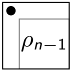
.

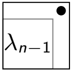
  - In particular, for such π, $Av(231, \pi)$ and $Av(132, P(\pi))$ are Wilf-equivalent, *i.e.* have the same enumeration.
  - For such π, regardless of k, the generating function of $Av(231, \pi)$ is F_n, where $F_1(z) = 1$ and $F_{n+1}(z) = \frac{1}{1-zF_n(z)}$.

Infinitely many equi-enumeration results from P

- **Theorem** [Albert & Bouvel 13]:
  - A pattern $\pi \in Av(231)$ is such that P provides a bijection between $Av(231, \pi)$ and $Av(132, P(\pi))$ if and only if

$\pi =$ 

, where $\lambda_1 = \rho_1 = \boxdot$, $\lambda_n =$ 

, $\rho_n =$ 

.

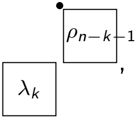
  - In particular, for such π, $Av(231, \pi)$ and $Av(132, P(\pi))$ are Wilf-equivalent, *i.e.* have the same enumeration.
  - For such π, regardless of k, the generating function of $Av(231, \pi)$ is F_n, where $F_1(z) = 1$ and $F_{n+1}(z) = \frac{1}{1 - zF_n(z)}$.

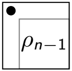
- **Consequence**:

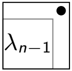
For each n, we obtain $2n$ Wilf-equivalent permutation classes.

Infinitely many equi-enumeration results from P

- Theorem [Albert & Bouvel 13]:
  - A pattern $\pi \in Av(231)$ is such that P provides a bijection between $Av(231, \pi)$ and $Av(132, P(\pi))$ if and only if

$\pi =$ 

, where $\lambda_1 = \rho_1 =$ 
, $\lambda_n =$ 

, $\rho_n =$ 

.

  - In particular, for such π, $Av(231, \pi)$ and $Av(132, P(\pi))$ are Wilf-equivalent, *i.e.* have the same enumeration.
  - For such π, regardless of k, the generating function of $Av(231, \pi)$ is F_n, where $F_1(z) = 1$ and $F_{n+1}(z) = \frac{1}{1 - zF_n(z)}$.

- Consequence:

For each n, we obtain $2n$ Wilf-equivalent permutation classes.

- Future work: Generalization to other graphically-guided bijections

Sorting with stacks and reverse

- **S** the stack-sorting operator
- **R** the reverse operator, defined by $\mathbf{R}(\sigma_1\sigma_2\ldots\sigma_n) = \sigma_n\ldots\sigma_2\sigma_1$.

• Question: Fix **A** any composition of **S** and **R**, like $\mathbf{A} = \mathbf{S} \circ \mathbf{R} \circ \mathbf{S} \circ \mathbf{S} \circ \mathbf{R} \circ \mathbf{S}$. Which permutations are sortable by **A**?

A	Characterization	Enumeration
$\mathbf{S}$	[Knuth 68]	[Knuth 68]
$\mathbf{S} \circ \mathbf{S}$	[West 93]	[Zeilberger 92]
$\mathbf{S} \circ \mathbf{R} \circ \mathbf{S}$ $\mathbf{S} \circ \alpha \circ \mathbf{S}$	[Albert, Atkinson, Bouvel, Claesson & Dukes 11]	[Bouvel & Guibert 12]
$\mathbf{S} \circ \mathbf{S} \circ \mathbf{S}$	[Úlfarsson 11]	??
More stacks	??	??

The original question of Claesson, Dukes, Steingrìmsson is about permutations sortable by stacks and symmetries α, among which **R**.

Sorting with stacks and reverse

- **S** the stack-sorting operator
- **R** the reverse operator, defined by $\mathbf{R}(\sigma_1\sigma_2\ldots\sigma_n) = \sigma_n\ldots\sigma_2\sigma_1$.

• Question: Fix **A** any composition of **S** and **R**, like $\mathbf{A} = \mathbf{S} \circ \mathbf{R} \circ \mathbf{S} \circ \mathbf{S} \circ \mathbf{R} \circ \mathbf{S}$. Which permutations are sortable by **A**?

• Theorem [Albert & Bouvel 13]:
For any operator **A** which is a composition of operators **S** and **R**, there are as many permutations of size n sortable by $\mathbf{S} \circ \mathbf{A}$ as permutations of size n sortable by $\mathbf{S} \circ \mathbf{R} \circ \mathbf{A}$.
Moreover, many permutation statistics are (jointly) equidistributed across these two sets.

Sorting with stacks and reverse

- **S** the stack-sorting operator
- **R** the reverse operator, defined by $\mathbf{R}(\sigma_1\sigma_2\ldots\sigma_n) = \sigma_n\ldots\sigma_2\sigma_1$.

• Question: Fix **A** any composition of **S** and **R**, like
$\mathbf{A} = \mathbf{S}\circ\mathbf{R}\circ\mathbf{S}\circ\mathbf{S}\circ\mathbf{R}\circ\mathbf{S}$. Which permutations are sortable by **A**?

• Theorem [Albert & Bouvel 13]:
For any operator **A** which is a composition of operators **S** and **R**, there are as many permutations of size n sortable by $\mathbf{S}\circ\mathbf{A}$ as permutations of size n sortable by $\mathbf{S}\circ\mathbf{R}\circ\mathbf{A}$.
Moreover, many permutation statistics are (jointly) equidistributed across these two sets.

The bijection P is the key to defining the bijection $\Psi_{\mathbf{A}}$ between $\mathbf{S}\circ\mathbf{A}$-sortable permutations and $\mathbf{S}\circ\mathbf{R}\circ\mathbf{A}$-sortable permutations.

Structure from substitution decomposition

Substitution decomposition of combinatorial objects

Analogue of the decomposition of integers as products of primes

- [Möhring & Radermacher 84]: general framework
- Applies to relations, graphs, posets, boolean functions, set systems, ...
- Permutations (almost) fit into this framework

Substitution decomposition of combinatorial objects

Analogue of the decomposition of integers as products of primes

- [Möhring & Radermacher 84]: general framework
- Applies to relations, graphs, posets, boolean functions, set systems, ...
- Permutations (almost) fit into this framework

Relies on:

- a principle for building objects (permutations, graphs) from smaller objects: the substitution
- some "basic objects" for this construction: **simple** permutations, **prime** graphs

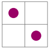
Required properties:

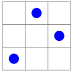
- every object can be decomposed using only "basic objects"
- this decomposition is unique

Substitution for permutations

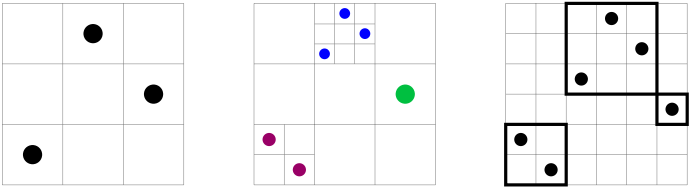
Substitution or inflation : $\sigma = \pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$.

Example: Here, $\pi = 1\,3\,2$, and

$\alpha^{(1)} = 2\,1 =$ 

$\alpha^{(2)} = 1\,3\,2 =$ 

$\alpha^{(3)} = 1 =$  .



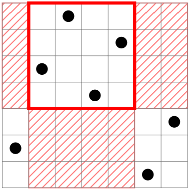
Hence $\sigma = 1\,3\,2[2\,1, 1\,3\,2, 1] = 2\,1\,4\,6\,5\,3$.

Simple permutations

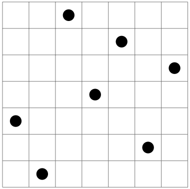
Interval (or block) = set of elements of σ whose positions **and** values form intervals of integers
Example: 5 7 4 6 is an interval of 2 5 7 4 6 1 3

Simple permutation = permutation with no interval, except the trivial ones: $1, 2, \ldots, n$ and σ
Example: 3 1 7 4 6 2 5 is simple

Not simple:



Simple:



Simple permutations

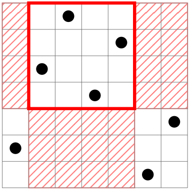
Interval (or block) = set of elements of σ whose positions **and** values form intervals of integers
Example: 5 7 4 6 is an interval of 2 5 7 4 6 1 3

Simple permutation = permutation with no interval, except the trivial ones: $1, 2, \ldots, n$ and σ
Example: 3 1 7 4 6 2 5 is simple

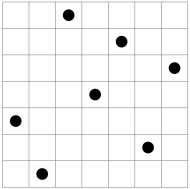
The smallest simple permutations:
$12, 21$, $2413, 3142$, 6 of size 5, ...
Remark:
It is convenient to consider 12 and 21 **not** simple.

Enumeration of simple permutations:
- Asymptotically $\frac{n!}{e^2}$, but no exact enumeration.
- The generating function is not D-finite.

Not simple:



Simple:



Substitution decomposition theorem for permutations

Theorem: [Albert, Atkinson & Klazar 03]
Every σ ($\neq 1$) is uniquely decomposed as

- $12 \ldots k[\alpha^{(1)}, \ldots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are $\oplus$-indecomposable
- $k \ldots 21[\alpha^{(1)}, \ldots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are $\ominus$-indecomposable
- $\pi[\alpha^{(1)}, \ldots, \alpha^{(k)}]$, where π is simple of size $k \geq 4$

Remarks:

- $\oplus$-indecomposable: that cannot be written as $12[\alpha^{(1)}, \alpha^{(2)}]$
- $\ominus$-indecomposable: that cannot be written as $21[\alpha^{(1)}, \alpha^{(2)}]$
- Allows to relate the generating function for simples with that of all permutations

Substitution decomposition theorem for permutations

Theorem: [Albert, Atkinson & Klazar 03]
Every σ ($\neq 1$) is uniquely decomposed as

- $12\ldots k[\alpha^{(1)},\ldots,\alpha^{(k)}]$, where the $\alpha^{(i)}$ are $\oplus$-indecomposable
- $k\ldots 21[\alpha^{(1)},\ldots,\alpha^{(k)}]$, where the $\alpha^{(i)}$ are $\ominus$-indecomposable
- $\pi[\alpha^{(1)},\ldots,\alpha^{(k)}]$, where π is simple of size $k \geq 4$

Remarks:

- $\oplus$-indecomposable: that cannot be written as $12[\alpha^{(1)},\alpha^{(2)}]$
- $\ominus$-indecomposable: that cannot be written as $21[\alpha^{(1)},\alpha^{(2)}]$
- Allows to relate the generating function for simples with that of all permutations

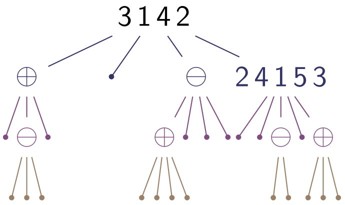
Decomposing recursively inside the $\alpha^{(i)}$ $\Rightarrow$ **decomposition tree**

[Flajolet & Sedgewick 09]: Trees are easy to study and enumerate.

Decomposition tree: witness of this decomposition

Example: Decomposition tree of

$\sigma =$ 10 13 12 11 14 1 18 19 20 21 17 16 15 4 8 3 2 9 5 6 7



Notations and properties:

- $\oplus = 12\ldots k$, $\ominus = k\ldots 21$ = linear nodes.
- π simple of size ≥ 4 = prime node.
- No edge $\oplus - \oplus$ nor $\ominus - \ominus$.
- Rooted ordered trees.
- These conditions characterize decomposition trees.

$\sigma = 3\,1\,4\,2[\oplus[1,\ominus[1,1,1],1],1,\ominus[\oplus[1,1,1,1],1,1,1],2\,4\,1\,5\,3[1,1,\ominus[1,1],1,\oplus[1,1,1]]]$

Bijection between permutations and their decomposition trees.

When the number of simple permutations in $\mathcal{C}$ is finite

- Theorem [Albert & Atkinson 05]:

If $\mathcal{C}$ contains a finite number of simple permutations, then $\mathcal{C}$ has a finite basis and an algebraic generating function.

The proof is constructive and uses substitution decomposition to provide a (possibly ambiguous) tree grammar describing $\mathcal{C}$.

When the number of simple permutations in $\mathcal{C}$ is finite

- Theorem [Albert & Atkinson 05]:

If $\mathcal{C}$ contains a finite number of simple permutations, then $\mathcal{C}$ has a finite basis and an algebraic generating function.

The proof is constructive and uses substitution decomposition to provide a (possibly ambiguous) tree grammar describing $\mathcal{C}$.

But... How to obtain an unambiguous grammar, *i.e.* a combinatorial specification? And how to do it automatically?

When the number of simple permutations in $\mathcal{C}$ is finite

• Theorem [Albert & Atkinson 05]:
If $\mathcal{C}$ contains a finite number of simple permutations, then $\mathcal{C}$ has a finite basis and an algebraic generating function.

The proof is constructive and uses substitution decomposition to provide a (possibly ambiguous) tree grammar describing $\mathcal{C}$.

But. . . How to obtain an unambiguous grammar, *i.e.* a combinatorial specification? And how to do it automatically?

• Theorem [Brignall, Ruškuc & Vatter 08]: It is decidable whether $\mathcal{C}$ given by its finite basis contains a finite number of simples.

The hard part of the proof is to decide (using automata theory) whether $\mathcal{C}$ contains finitely many proper pin-permutations.

When the number of simple permutations in $\mathcal{C}$ is finite

• Theorem [Albert & Atkinson 05]:
If $\mathcal{C}$ contains a finite number of simple permutations, then $\mathcal{C}$ has a finite basis and an algebraic generating function.

The proof is constructive and uses substitution decomposition to provide a (possibly ambiguous) tree grammar describing $\mathcal{C}$.

But... How to obtain an unambiguous grammar, *i.e.* a combinatorial specification? And how to do it automatically?

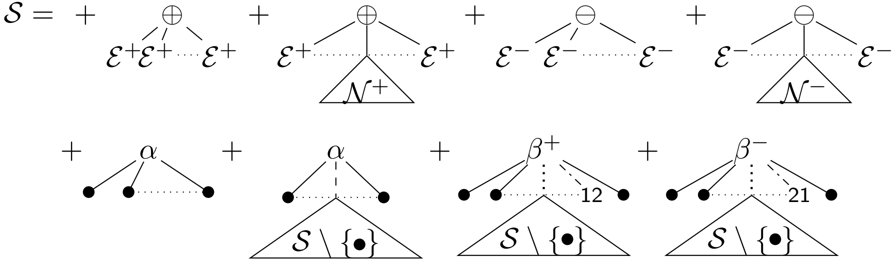
• Theorem [Brignall, Ruškuc & Vatter 08]: It is decidable whether $\mathcal{C}$ given by its finite basis contains a finite number of simples.

The hard part of the proof is to decide (using automata theory) whether $\mathcal{C}$ contains finitely many proper pin-permutations.

But... How to decide by an efficient algorithm whether $\mathcal{C}$ contains finitely many simples?

Characterization and enumeration of pin-permutations

- Characterization of the decomposition trees of pin-permutations:

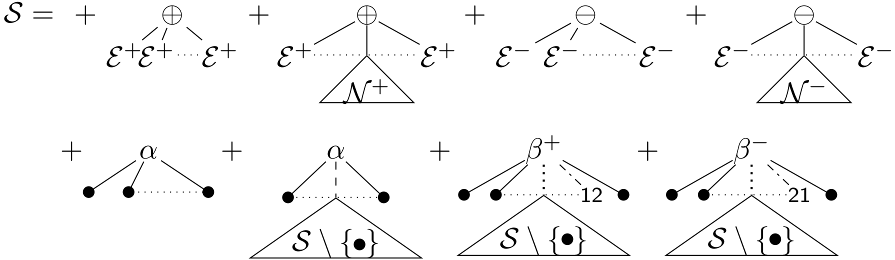


- Computation of the (rational) generating function of pin-permutations:

$$S(z) = z\frac{8z^6 - 20z^5 - 4z^4 + 12z^3 - 9z^2 + 6z - 1}{8z^8 - 20z^7 + 8z^6 + 12z^5 - 14z^4 + 26z^3 - 19z^2 + 8z - 1}$$

[Bassino, Bouvel & Rossin 11]

- Characterization of the decomposition trees of pin-permutations:



- Computation of the (rational) generating function of pin-permutations:

$$S(z) = z \frac{8z^6 - 20z^5 - 4z^4 + 12z^3 - 9z^2 + 6z - 1}{8z^8 - 20z^7 + 8z^6 + 12z^5 - 14z^4 + 26z^3 - 19z^2 + 8z - 1}$$

[Bassino, Bouvel & Rossin 11]

This is also a specific result obtained with substitution decomposition.

Computation of specifications of permutation classes

• Algorithm testing whether $\mathcal{C}$ given by its finite basis contains a finite number of simples. [Bassino, Bouvel, Pierrot & Rossin 13+]

- Based on substitution decomposition, our study of pin-permutations and automata theory.
- Complexity $\mathcal{O}(n \log n + n + s^{2k})$, to be compared to $\mathcal{O}(n \log n + n \cdot 8^{s'} + 2^{k \cdot s \cdot 2^s})$ for [BRV 08].

Computation of specifications of permutation classes

- Algorithm testing whether $\mathcal{C}$ given by its finite basis contains a finite number of simples. [Bassino, Bouvel, Pierrot & Rossin 13+]
  - Based on substitution decomposition, our study of pin-permutations and automata theory.
  - Complexity $\mathcal{O}(n \log n + n + s^{2k})$, to be compared to $\mathcal{O}(n \log n + n \cdot 8^{s'} + 2^{k \cdot s \cdot 2^s})$ for [BRV 08].
- Algorithm computing the set of simples $\mathcal{C}$, in case it is finite.
  - Analyzing the poset of simple permutations. [Pierrot & Rossin 12]

Computation of specifications of permutation classes

- Algorithm testing whether $\mathcal{C}$ given by its finite basis contains a finite number of simples. [Bassino, Bouvel, Pierrot & Rossin 13+]
  - Based on substitution decomposition, our study of pin-permutations and automata theory.
  - Complexity $\mathcal{O}(n \log n + n + s^{2k})$, to be compared to $\mathcal{O}(n \log n + n \cdot 8^{s'} + 2^{k \cdot s \cdot 2^s})$ for [BRV 08].
- Algorithm computing the set of simples $\mathcal{C}$, in case it is finite.
  - Analyzing the poset of simple permutations. [Pierrot & Rossin 12]
- Algorithm computing, from the finite set of simples in $\mathcal{C}$, a combinatorial specification for $\mathcal{C}$. [Bassino, Bouvel, Pierrot, Pivoteau & Rossin 12]
  - Propagate pattern avoidance/containment constraints into substitution decomposition.
  - Unlike [AA 05], algorithm computing non-ambiguous grammars.

Byproducts of specifications

$\Rightarrow$ Algorithmic chain
from a finite basis B to a combinatorial specification for $\mathcal{C} = Av(B)$

Byproducts of specifications

$\Rightarrow$ Algorithmic chain
from a finite basis B to a combinatorial specification for $\mathcal{C} = Av(B)$

From a combinatorial specification for $\mathcal{C}$, we immediately get:

- A polynomial system for the generating function $C(z)$
  [Flajolet & Sedgewick 09]
- Efficient random samplers of permutations in $\mathcal{C}$ (recursive or Boltzmann method) [Flajolet, Zimmerman & Van Cutsem 94] [Duchon, Flajolet, Louchard & Schaeffer 04]

Byproducts of specifications

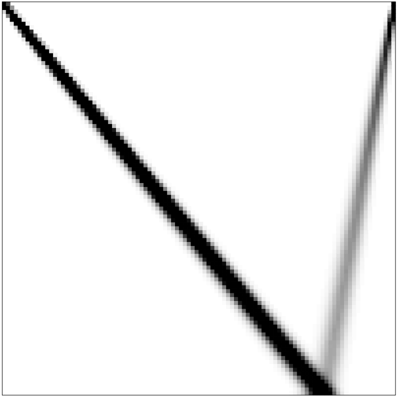
$\Rightarrow$ Algorithmic chain
from a finite basis B to a combinatorial specification for $\mathcal{C} = Av(B)$

From a combinatorial specification for $\mathcal{C}$, we immediately get:

- A polynomial system for the generating function $C(z)$
  [Flajolet & Sedgewick 09]
- Efficient random samplers of permutations in $\mathcal{C}$ (recursive or Boltzmann method)
  [Flajolet, Zimmerman & Van Cutsem 94]
  [Duchon, Flajolet, Louchard & Schaeffer 04]

$\Rightarrow$ Observation of many large random permutations in permutation classes

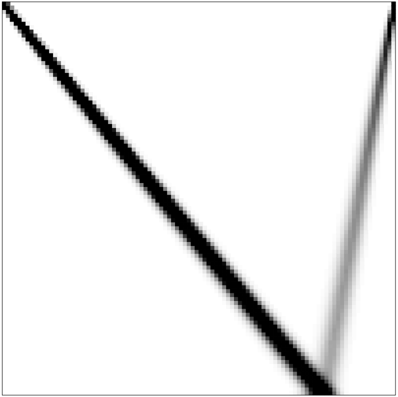
Example:
30 000 permutations of size 500 in $Av(2413, 1243, 2341, 531642, 41352)$



Study average properties of random permutations in permutation classes.

Asymptotic properties of permutations in classes

Example:
30 000 permutations of size 500 in $Av(2413, 1243, 2341, 531642, 41352)$



Study average properties of random permutations in permutation classes.

In the literature, only $Av(123)$ and $Av(132)$ have been studied from this perspective. [Miner & Pak 13]

Structure from graphs

Permutation patterns and induced subgraphs

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

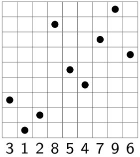
σ_i •—• σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

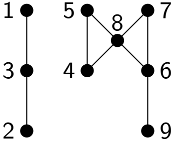
Permutation patterns and induced subgraphs

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

$\sigma_i \bullet\!\!-\!\!\!-\!\!\bullet \sigma_j$ is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

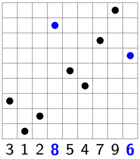
$\longmapsto$ $G_\sigma =$ 

Permutation patterns and induced subgraphs

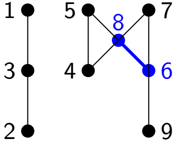
To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i •—• σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

$\longmapsto$

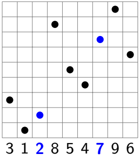
$G_\sigma =$ 

Permutation patterns and induced subgraphs

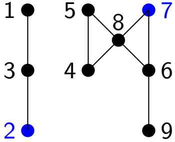
To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i •—• σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

$\longmapsto$

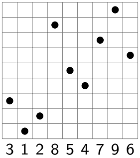
$G_\sigma =$ 

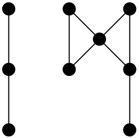
Permutation patterns and induced subgraphs

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i •—• σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

$\longmapsto$ $G_\sigma =$ 

Considering the unlabeled version of G_σ, this application is neither injective nor surjective.

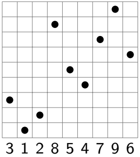
But permutation patterns correspond to induced subgraphs.

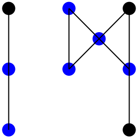
Permutation patterns and induced subgraphs

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i —— σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

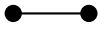
$\longmapsto$ $G_\sigma =$ 

Considering the unlabeled version of G_σ, this application is neither injective nor surjective.

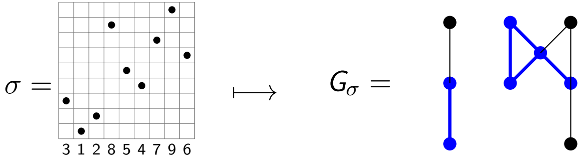
But permutation patterns correspond to induced subgraphs.

Permutation patterns and induced subgraphs

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i  σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:



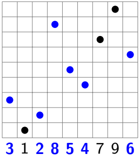
Considering the unlabeled version of G_σ, this application is neither injective nor surjective.

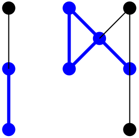
But permutation patterns correspond to induced subgraphs.

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i •—• σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

$\longmapsto$ $G_\sigma =$ 

Considering the unlabeled version of G_σ, this application is neither injective nor surjective.

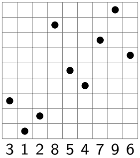
But permutation patterns correspond to induced subgraphs.

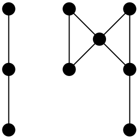
Permutation patterns and induced subgraphs

To $\sigma \in \mathfrak{S}$, associate the graph G_σ of the inversions of σ:

σ_i •—• σ_j is an edge of G_σ iff $i < j$ and $\sigma_i > \sigma_j$

Example:

$\sigma =$ 

$\longmapsto$ $G_\sigma =$ 

Considering the unlabeled version of G_σ, this application is neither injective nor surjective.

But permutation patterns correspond to induced subgraphs.

And permutation classes are the analogues of induced subgraph ideals (= sets of graphs that are downward closed when taking induced subgraphs).

From graphs to permutations and conversely

• The study of induced subgraph ideals (= **is** ideals) is a recent topic in graph theory. [Chudnovsky, Seymour and collaborators]

• Most results are of the form:
An **is** ideal $\mathcal{I}$ is such that a parameter (e.g. maximum degree) is bounded if and only if $\mathcal{I}$ does not include simpler **is** ideals (e.g. ideals of cliques and stars).

What can we obtain transposing this approach to permutation classes?

From graphs to permutations and conversely

- The study of induced subgraph ideals (= **is** ideals) is a recent topic in graph theory. [Chudnovsky, Seymour and collaborators]

- Most results are of the form:

An **is** ideal $\mathcal{I}$ is such that a parameter (e.g. maximum degree) is bounded if and only if $\mathcal{I}$ does not include simpler **is** ideals (e.g. ideals of cliques and stars).

What can we obtain transposing this approach to permutation classes?

- State permutation analogues of conjectures on induced subgraphs (and hopefully prove them).

Does it provide insight on the graph conjecture?

Erdős-Hajnal conjecture: For every graph H, there exists a constant $\delta(H) > 0$ such that every graph G with no induced subgraph isomorphic to H has either a clique or a stable set of size at least $|V(G)|^{\delta(H)}$.

Some perspectives

• From graphically-guided bijections, find infinities of Wilf-equivalences (= equi-enumeration results) between permutation classes.

This would also provide a unified framework for many known Wilf-equivalences.

• From combinatorial specifications obtained from substitution decomposition, study random permutations in permutation classes.

• Develop new problematics on permutation classes, inspired from those on induced subgraph ideals.

Some perspectives

• From graphically-guided bijections, find infinities of Wilf-equivalences (= equi-enumeration results) between permutation classes.

This would also provide a unified framework for many known Wilf-equivalences.

• From combinatorial specifications obtained from substitution decomposition, study random permutations in permutation classes.

• Develop new problematics on permutation classes, inspired from those on induced subgraph ideals.

Merci !